AMENDMENTS TO THE DRAWINGS

A replacement sheet containing Fig. 3 is attached. Fig. 3 has been amended to indicate the switching of modulus status bits (MSB) 32 from value "0" to value "1" (referenced by reference numeral 100). Also, a "COMPARE" box 102 has been added to indicate comparison of the MSB 32 with the MSB of an issuing instruction. A "MODULUS OPERATION" box 104 is added to illustrate a modulus operation performed on the LSN (load store number) of load instructions.

No new matter has been added by this amendment.

REMARKS

In the Office Action dated September 22, 2005, the drawings were objected to; claims 7, 8, 17, and 18 were rejected under 35 U.S.C. § 112, ¶ 1; and claims 1-4, 7, 8, 10, 12-14, 17, 18, and 20-31 were rejected under § 103 over U.S. Patent No. 6,182,210 (Akkary) in view of "The Metaflow Architecture," IEEE Micro (1991) (Popescu).

TERMINAL DISCLAIMER

The previous terminal disclaimer, filed on June 30, 2005, was indicated as being accepted and recorded. However, Applicant has discovered that a power of attorney making L. Joy Griebenow (the attorney who signed the terminal disclaimer) an attorney of record was not filed prior to submission of the terminal disclaimer. To remedy this inadvertent oversight, a new power of attorney was filed on November 8, 2005.

Also, a replacement terminal disclaimer (identical in substance to the previously submitted terminal disclaimer) is submitted herewith, along with a check enclosing the required terminal disclaimer fee.

The previously submitted terminal disclaimer, filed June 30, 2005, is hereby withdrawn in favor of the presently submitted terminal disclaimer.

OBJECTIONS TO DRAWINGS

Fig. 3 has been amended to address the drawing objections, which stated that the features of claims 8 and 25-28 must be shown. Claim 8 recites switching the status bit of the selected location when the selected location becomes invalid. In Fig. 3, arrows point from "0" to "1" to illustrate the switching of the value of each MSB 32. Support for switching the MSB value can be found at least in the following passages of the present specification: page 5, lines 3-7; page 9, lines 5-8.

Claim 25 recites that the selected location is determined to be unavailable in response to determining that the stored status bit associated with the selected location does not match the calculated status bit. A "COMPARE" box 102 is added to Fig. 3 to illustrate the comparison of the stored MSB value 32 and the MSB of the issuing instruction. Support for this amendment of Fig. 3 can be found at least in the following passage of the specification: page 4, line 22-page 5, line 2.

Claim 26 recites changing a state of the stored status bit of each of the plurality of locations of the execution queue in response to completion of all program steps stored in the plurality of locations. Claim 26 is now supported by the change to Fig. 3 that illustrates the switching of the MSB values 32 from "0" to "1." Support for this amendment of Fig. 3 can be found in the following passages of the specification: page 5, lines 3-7; page 9, lines 5-8.

Claim 27 recites means for performing a modulus operation on the unique number of the issued program step to calculate a queue entry number that selects the specific one of the plurality of locations in the execution queue. A "MODULUS OPERATION" box 104 is added to Fig. 3 to illustrate the modulus operation performed on the LSN (load store number) of a load instruction. Support for this amendment is found in the specification on page 7, at lines 19-21.

Claim 28 recites performing the modulus operation that also produces the value of the status bit. The modulus operation represented by box 104 in Fig. 3 produces such a status bit.

No new matter has been added as a result of the amendments of Fig. 3. In view of the amendment of Fig. 3, it is respectfully submitted that the objections to the drawings has been overcome.

SUPPORT FOR AMENDMENT TO THE SPECIFICATION

The paragraph on 8, starting at line 22, has been amended by adding two sentences that were substantially copied from the "Summary of the Invention" section of the present specification on page 5, at lines 3-7.

The new paragraph inserted on page 9 was substantially copied from the "Summary of the Invention" section on page 4, line 22 to page 5, line 2.

Since support for the amendments to the specification can be found in the original disclosure, no new matter has been added.

REJECTION UNDER 35 U.S.C. § 112, ¶ 1

Claims 7 and 17 have been amended to address the § 112 rejection. Withdrawal of the § 112 rejection is therefore respectfully requested.

REJECTION UNDER 35 U.S.C. § 103

The obviousness rejection of claim 31 is addressed first. The Office Action identified component 104 (instruction cache) of Akkary as being the first queue recited in claim 31. 9/22/2005 Office Action at 12. The Office Action also identified the load buffer 182A depicted in Fig. 24 of Akkary as being the execution queue recited in claim 31. *Id.*

The "first queue" element of claim 31 recites that the first queue is to store program steps in a program order, where the program steps are assigned respective program numbers that correspond to the program order. The Office Action identified the LBID (load buffer ID) assigned to load instructions as being the program numbers recited in claim 31. *Id.* The Office Action also stated that the instructions "are fetched in order but [are] issued out of order." *Id.*

Fig. 24 shows load instructions "load 0," "load 1," and "load 2" stored in an order that is specified by LBID 0, 1, and 2. Column 20 of Akkary shows a program order of several store and load instructions, in which "load 0" comes before "load 1" which comes before "load 2." Thus, it is apparent that the load buffer 182A depicted in Fig. 24 of Akkary stores load instructions in the same order as the program order for these load instructions. Note that the load buffer 182A is part of the memory order buffer (MOB) 178, which holds copies of load and store instructions of traces in trace buffers 114. Akkary, 17:63-65. The trace buffers 114 also hold instructions in program order. Akkary, 16:43-44. Significantly, it is noted that the trace buffer dispatches instructions for execution in the order the instructions exist in the trace buffer, which is the *program order.* Akkary, 16:41-44. Therefore, it is clear that in Akkary, the trace buffer dispatches instructions to the MOB (including load buffers) in *program order.* This is inconsistent with the recitation in claim 31 that states that the first queue transfers at least some of the program steps to the execution queue *out of the program order.*

In view of the mis-application of Akkary to the "first queue" and "execution queue" elements of claim 31, it is respectfully submitted that a *prima facie* case of obviousness has not been established with respect to claim 31 over the asserted combination of Akkary and Popescu.

The *prima facie* case of obviousness is further defective for the reason that the hypothetical combination of Akkary and Popescu fails to teach or suggest that locations of the execution queue store respective program steps based on queue entry numbers calculated from numeric operations on the program numbers of respective program steps. The Office Action

conceded that Akkary does not disclose this element. 9/22/2005 Office Action at 12. However, the Office Action stated that Popescu teaches "that the index position of the instruction in a queue wraps around from its maximum value back to its minimum value, and using a 'color' bit to determine where the older instructions start by flipping the color bit when the index of the queue wraps around ...." *Id.* at 12-13. The Office Action further stated that "[i]n order for an index to wrap around, there must inherently be a modulus operation of the location number occurring ...." *Id.* at 13. This explanation of Popescu does not address the specific words of claim 31, namely that the queue entry numbers are calculated from numeric operations *on the program numbers* of respective program steps. The Office Action refers to a "modulus operation of the location number" occurring in Popescu and a "modulus calculation ... based on the total number of locations available." *Id.* at 13. Performing the modulus operation on the location number or a modulus calculation on the total number of locations available, even if they were occurring in Popescu, is *not* the same as the subject matter recited in claim 31, namely that the locations of the execution queue to store respective program steps are based on queue entry numbers calculated from numeric operations *on the program numbers* of respective program steps. Therefore, since neither Akkary nor Popescu discloses or even remotely suggests this element of claim 31, a *prima facie* case of obviousness clearly has not been established.

In addition, the hypothetical combination of Akkary and Popescu also fails to disclose or suggest the first queue to further determine whether a particular location of the execution queue is available by calculating a status bit based on the program number of the respective program step, and comparing the calculated status bit with the stored status bit to determine whether the particular location is available. The Office Action equated the status bit with the "color bit" disclosed by Popescu. There clearly is no teaching or suggestion by Popescu that the color bit of Popescu is used for determining whether a particular location is available. The specific teaching of Popescu is that the color bit is toggled when an index portion wraps around, and that the color bit "simplifies the process of determining the *relative age* of two shelved instructions." Popescu, at 12 (emphasis added). Thus, Popescu specifically teaches that the color bit is used to determine which instruction is *older*. There is no teaching or suggestion that this color bit is used to determine whether the DRIS is full.

Moreover, the comparison of color bits performed in Popescu is color bits of two different *shelved* instructions. In other words, the comparison of color bits is occurring for instructions that are already *stored* in the DRIS. This is a clear indication that the color bit is *not* used to determine whether a location of the DRIS is available. Therefore, since the hypothetical combination of Akkary and Popescu does not teach this additional element, a *prima facie* case of obviousness has not been established for this additional reason.

Also, there was no suggestion or motivation to combine the teachings of Akkary and Popescu. More specifically, there was no motivation or suggestion to combine the deferred-scheduling, register-renaming instruction shelf (DRIS) of Popescu with the load buffer of Akkary. Popescu teaches the concept of instruction shelving, where out-of-order execution is used to overcome the performance loss caused by stalling. Popescu, at 12. "The principle of out-of-order execution is that the issue logic shelves any stalled instruction and continues to issue subsequent instructions, returning to the shelved instruction only after all its operands are available." *Id.* The shelved instructions are placed in the DRIS, as depicted on page 63 of Popescu. The Office Action cited to paragraph 4 on page 64 of Popescu as purportedly teaching the elements missing from Akkary. The cited passage of Popescu refers to allocating an identifier (ID) to each shelved instruction, where the ID consists of an index into the DRIS, plus a "color" bit. The IDs of Popescu are allocated in "strict numerical order," with the color bit toggled when the index portion wraps around. The color bit of Popescu is used to determine the relative age of two shelved instructions in the DRIS.

The teaching of Popescu with respect to shelved instructions, which are stalled instructions, stored in the DRIS is completely unrelated to the load buffer, used for storing load instructions, as taught by Akkary. A person of ordinary skill in the art looking to the disparate teachings of Akkary and Popescu would not have been motivated to combine the teachings regarding the DRIS in Popescu with the teachings regarding the load buffer of Akkary.

Except for the disclosure of the present invention, there was no motivation to combine unrelated elements of Akkary and Popescu in the manner proposed by the Office Action. Thus, the Office Action has engaged in the use of impermissible hindsight to piece together unrelated prior art reference teachings when no suggestion or motivation existed to make such a combination. The *prima facie* case of obviousness is defective for this additional reason.

Independent claim 1 is also allowable over the asserted combination of Akkary and Popescu. Note that claim 1 recites that each of a plurality of program steps is assigned a unique number, where the numbers indicate a program order of the program steps. Also, claim 1 recites issuing a program step to an execution queue out of the program order. As discussed above, the trace buffer 114 of Akkary dispatches instructions to the load queue 182A depicted in Fig. 24 in program order. Therefore, instructions are not issued from the trace buffer 114 to the load buffer 182A out of program order.

This point is further emphasized by the addition of the follow clause of claim 1: "wherein selecting the location in the execution queue enables the issued program step to be maintained in the program order in the execution queue even though the issued program step is issued to the execution queue out of the program order."

Therefore, it is clear that the hypothetical combination of Akkary and Popescu does not teach or suggest all elements of claim 1, and thus a *prima facie* case of obviousness cannot be established with respect to claim 1.

Moreover, the color bit of Popescu, equated to the status bit of claim 1 by the Office Action, is used to indicate the relative age of an instruction in the DRIS. This color bit is not used to determine whether a selected location is available. Claim 1 expressly recites: based upon the determined number of the instruction valid bit and the calculated value of the status bit, determining availability of the selected location. However, as explained above, the color bit of Popescu clearly is not used to perform such a determination of availability of a selected location of the execution queue.

For this additional reason, a *prima facie* case of obviousness has not been established.

Moreover, as explained above, no motivation or suggestion existed to combine the teachings of Akkary and Popescu, which is a further reason that no motivation or suggestion existed to combine reference teachings.

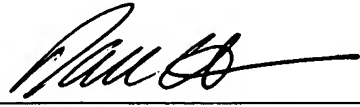Independent claim 10 is allowable for similar reasons as claim 1.

Dependent claims are allowable for at least the same reasons as corresponding independent claims.

In view of the foregoing, allowance of all claims is respectfully requested. The Commissioner is authorized to charge any additional fees and/or credit any overpayment to Deposit Account No. 08-2025 (200304950-2).

Respectfully submitted,

Date:_____Nov, 22, 2005_____   _____

Dan C. Hu
Registration No. 40,025
TROP, PRUNER & HU, P.C.
8554 Katy Freeway, Suite 100
Houston, TX 77024
Telephone: (713) 468-8880
Facsimile: (713) 468-8883